Lots of discussion at COMP meetings

- Jet class, parton class, vertex class
- Hydro wrappers almost done
- All on GitHub COMP
- Two basic framework setups have been proposed
  - Somewhat different methods of interfacing between modules

Lots of discussion at COMP meetings

- Have started evaluating the modularity of a couple different approaches
  - One that Joern has proposed
  - One that Dragos has proposed
- Will evaluate how easy it is to isolate changes to
  - The Jet Class/Code
  - The Hydro Class/Code

Lots of discussion at COMP meetings

- Discussion regarding acceleration on GPU and Xeon Phi
- Consensus moving toward supporting
  - CUDA
  - OpenCL
  - Focus on the most computationally intensive functions

Coding standard

- Use C++11 which is smarter and easier
- Use CMake to compile the program
  - independent of platforms
  - look for library path automatically
- Code restructuring (contributors should learn new skills and google coding standard to make a uniform style).

Contributors

- Hydrodynamic interface: Chun Shen, LongGang Pang
- JetClass interface: Abhijit, ShanShan Cao
- Pythia interface: richard-cms
- New Framework: Joern Putschke

Ask for more contributors

- Documents, make->cmake, UnitTests, Configuration file reading module, Trento initial condition

Become a new contributor

1. Create GitHub account, install git in local machine
2. Ask for privilege from Abhijit for JETSCAPE-COMP repository
3. “Fork” a copy of repository JETSCAPE-COMP from Abhijit
4. Clone the code to local machine in command line:

    git clone https://username@github.com/username/JETSCAPE-COMP.git

5. Do modification, commit, push to your github server
6. Go to Abhijit’s account, do “pull request” to ask others to review your code and request Abhijit to merge your contribution